2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

Multivariate Time Series Prediction based on Multiple Kernel Extreme Learning Machine

Xinying Wang and Min Han

***Abstract*— In this paper, a multiple kernel extreme learning machine (MKELM) is proposed for multivariate time series prediction. The multivariate time series is reconstructed in phase space, and a variable selection algorithm is then applied to form the compact and relevant input for the prediction model. On the basis of multiple kernel learning and extreme learning machine with kernels, multi different kernels is used in MKELM to present the dynamics of multivariate time series. A simulation example, prediction of Lorenz chaotic time series is conducted to demonstrate the effectiveness of the proposed method.**

I. Introduction

DURING the last decades, time series prediction has played an important role in both science research and engineering applications [1], [2]. Commonly, time series has the characteristic of nonlinearity, and consists of multiple variables [3]. However, most of the published papers are concerned with univariate time series other than the multivariate time series. However, multivariate time series often contains more dynamic information of the underlying system than univariate time series [4]. As a result, the research on multivariate time series prediction has drawn an increasing focus [5], [6].

Neural networks, which have a strong nonlinear mapping ability, have been one of the most influential prediction tools. According to the Takens' delay embedding theorem [7], the time series can be reconstructed to the phase space by the delayed coordinate, translating the time correlation to the spatial correlation. With the universal approximation capability, neural networks can be an effective prediction model. But the traditional gradient-based learning algorithms of neural networks convergence slow and are easy to be trapped in local optimum, which has constrained the further application of neural networks in the field of time series prediction.

In order to overcome the shortcomings of traditional neural networks, extreme learning machine (ELM) [8] is proposed. The input weights and the hidden layer biases of ELM are randomly generated and keep fixed during the learning progress. Only the output weights need to be tuned and linear regression can obtain satisfying results. As a result, ELM has been successfully applied to time series prediction [9], [10], [11]. Although the ELM model has greatly improved the neural network training speed and accuracy, there are also some shortcomings of ELM itself, i.e. the output weights calculation process is an ill-posed problem, the optimal structure of the ELM is hard to be determined. Combining ELM with support vector machines (SVM) by replacing the hidden layer mapping of ELM with kernel function mapping of SVM, extreme learning machine with kernels (KELM) [12] is developed. KELM model avoids the optimal structure determination problem and retains the advantage of fast training speed of ELM. However, it is the single kernel used in KELM that leads to some shortcomings of single kernel methods such as limited dynamic representation capability and complex parameter optimization.

At the same time, considering the different dynamic features of multivariate time series, it is suggested that for multivariate time series prediction problem the adaptation of a single predictor might not be enough [13], [14], [15]. In this paper, a prediction model based on multiple kernel extreme learning machine (MKELM) is proposed. Inspired by the multiple kernel learning (MKL), which uses hybrid kernels to obtain more dynamic information in the feature space [16], [17], ELMK is extended to use multiple different kernels to represent the dynamics of multivariate time series.

II. Extreme Learning Machine with Kernels

Mathematically, ELM [18] can be formulated as follows

$$\sum_{i=1}^{L} w_i g(x_j) = \sum_{i=1}^{L} w_i g(W_{in(i)} \cdot x_j + b_i) = y_j, j = 1, ..., N. \tag{1}$$

where $x_j \in \Re^n$ is the input vector, $W_{in(i)} \in \Re^n$ is the weight vector connecting the input nodes to the $i-$th hidden node, $W_{in(i)} \cdot x_j$ denotes the inner product of $W_{in(i)}$ and x_j, $b_i \in \Re$ is the bias of the $i-$th hidden node, $g(\cdot)$ is the sigmoid activation function , $w_i \in \Re$ is the weight connecting the $i-$th hidden node to the output node, $y_j \in \Re$ is the output of ELM, L is the number of the hidden nodes and N is the number of training samples.

Let $T = [t_1, ..., t_N]^{\mathrm{T}}$, $w = [w_1, w_2, \cdots, w_L]^{\mathrm{T}}$ and

$$H = \begin{pmatrix} g(W_{in(1)}, b_1, x_1) & \cdots & g(W_{in(L)}, b_L, x_1) \\ \vdots & \ddots & \vdots \\ g(W_{in(1)}, b_1, x_N) & \cdots & g(W_{in(L)}, b_L, x_N) \end{pmatrix}_{N \times L}$$

where T is the desired vector and matrix H is called the hidden layer output matrix of ELM; the ith column of H is the ith hidden node's output vector with respect to inputs $x_1, x_2, ..., x_N$ and the jth row of H is the output vector of the

Xinying Wang and Min Han are with the Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian, Liaoning, China. (email: minhan@dlut.edu.cn).

This work was supported by National Natural Science Foundation of China under Grant Nos. 61374154 and the National Basic Research Program of China under Grant Nos. 2013CB430403.

978-1-4799-1484-5/14/$31.00 ©2014 IEEE

hidden layer with respect to input x_j. If the ELM model with L hidden nodes can learn these N training samples with no residuals, then it means that there exist w so that

$$Hw = T \tag{2}$$

The least squares solution of (2) is

$$w = H^{\dagger}T \tag{3}$$

where $H^{\dagger}$ is the Moore-Penrose generalized inverse of matrix H.

The training process of ELM is a simple linear regression, which can effectively overcome the inherent flaws of traditional neural networks. However, the number of hidden layer nodes, which is an important parameter of ELM crucial to the performance of prediction model, usually should be selected by some time-consuming methods according to the learning tasks [19], [20]. Avoiding the hidden nodes selection problem, extreme learning machine with kernels (ELMK) [12] is developed, by replacing the hidden layer mapping $h(x)$ in ELM by the kernel function mapping $\phi(x)$ in SVM. Consequently, the replaced hidden layer mapping can be unknown. As a result, the kernel matrix of ELM can be defined as follows

$$\begin{aligned}&\Omega_{ELM} = HH^T:\\ &\Omega_{ELMi,j} = h(x_i)\cdot h(x_j) = K(x_i \cdot x_j)\end{aligned}$$

The output function can be written as

$$\begin{aligned}f(x) &= h(x)H^T\left(\tfrac{I}{C} + HH^T\right)^{-1}T\\ &= \begin{bmatrix} K(x\cdot x_1)\\ \vdots \\ K(x\cdot x_N)\end{bmatrix}^T\left(\tfrac{I}{C}+\Omega_{ELM}\right)^{-1}T\end{aligned}$$

The hidden layer mapping in the special kernel implementation of ELM can be unknown, but the corresponding kernel is usually given. Therefore, there is no longer need to identify the number of the hidden nodes (the structure of ELM) [12].

Given a training set $T = (x_i, t_i), i = 1, \ldots, N$, where $x_i \in \Re^P$, and $t_i \in \Re$. The original optimization problem of ELMK can be written as

$$\begin{aligned}&\min L_P = \tfrac{1}{2}\|w\|^2 + C\tfrac{1}{2}\sum_{i=1}^{N}\xi_i^2\\ &s.t. \quad \phi(x_i)\cdot w = t_i - \xi_i, 0, i = 1, \ldots, N\end{aligned} \tag{4}$$

where w is a vector in the feature space $\mathbf{F}$, and $\phi(x)$ maps the input x to a vector in $\mathbf{F}$. C is the regularization parameter. Here, we use $\phi(x)$ instead of $h(x)$ in order to keep consistent with support vector machine and implicitly indicate that the mapping is unknown. ξ is the error.

The corresponding Lagrangian dual problem can be formated as

$$L_D = \frac{1}{2}\|w\|^2 + C\frac{1}{2}\sum_{i=1}^{l}\xi_i^2 - \sum_{i=1}^{l}\theta_i\left(\phi(x_i)w - t_i + \xi_i\right) \tag{5}$$

III. Multiple Kernel Extreme Learning Machine

Based on the aforementioned analysis, multiple kernel extreme learning machine (MKELM) is proposed in this paper. The reasoning is similar to combining different predictors: instead of choosing a single optimal ELMK predictor and putting all eggs in the same basket, it is better to have a set and let an algorithm do the picking or combination [21], and a single ELMK predictor may not efficient enough to model the multivariate time series [14], [15]. There can be two uses of MKELM: (a) Different kernels correspond to different time scales dynamics, using a combination of kernels can express multi time scale system dynamics. (b) Different kernels may be using inputs coming from different time series possibly from different sources or locations.

In MKELM, the kernel $K(x, x')$ is actually a convex linear combination of other single ELMK kernels

$$\begin{aligned}K(x,x') &= \sum_{k=1}^{M}\mu_k K_k(x,x')\\ s.t. \quad &\mu_k \geq 0, \sum_{k=1}^{M}\mu_k = 1\end{aligned} \tag{6}$$

where M is the total number of kernels, which corresponding to signal ELMK. Thus, the optimization problem of MKELM can be written as

$$\begin{aligned}&\min L_{MKELM} = \tfrac{1}{2}\textstyle\sum_k \frac{1}{\mu_k}\|w_k\|^2 + C\frac{1}{2}\sum_{i=1}^{N}\xi_i^2\\ &s.t. \quad \textstyle\sum_k \phi_k(x_i)w_k = t_i - \xi_i, i = 1, \ldots, N\\ &\quad\ \textstyle\sum_k \frac{1}{\mu_k} = 1, k = 1, \ldots, M\end{aligned} \tag{7}$$

The approach for solving the MKELM optimization problem is to use a 2-step alternate optimization algorithm [22]. While considering that the vector μ is fixed, the first step would consist in solving the dual problem

$$J(\mu) = -\frac{1}{2}\sum\nolimits_{i,j}\alpha_i\alpha_j\sum_{k=1}^{M}\mu_k K_k(x,x') - \frac{1}{2C}\sum\nolimits_i\alpha_i^2 \tag{8}$$

This is the usual ELMK dual problem using a single hybrid kernel matrix $K(x_i, x_j) = \sum_k \mu_k K_k(x_i, x_j)$. Hence, this step can be solved by the algorithm proposed in [12]. The more efficient the ELMK algorithm is, the more efficient the MKELM algorithm becomes. The overall complexity of MKELM algorithm is tied to the one of the single kernel ELMK algorithm.

Then, the second step consists in updating the weight vector μ. For a given μ, $J(\mu)$ is the objective value of original ELMK where the kernel K is the aggregation of each individual kernel weighted by μ. Therefore, $J(\mu)$ is convex, which ensures global convergence, and differentiable on μ [17]. The differentiation of $J(\mu)$ with respect to μ_k can be written as

$$\frac{\partial J}{\partial \mu_k} = -\frac{1}{2}\sum\nolimits_{i,j}\alpha_i\alpha_j K_k(x,x') \tag{9}$$

Finally, the algorithm of MKELM can be summarized as

1: **initialization**

Set weight $\mu_k = \frac{1}{M}, k = 1, \ldots, M$
while stopping criteria is not reached **do**
  Solve the original ELMK J with the combined kernel $K = \sum_k \mu_k K_k$
  Compute gradient descent D of J w.r.t. μ_k for $k = 1, \ldots, M$
  Set $d = \arg\max_k \mu_k, J^\dagger = 0, \mu^\dagger = \mu, D^\dagger = D$
  while $J^\dagger < J(\mu)$ **do**
    $\mu = \mu^\dagger, D = D^\dagger$
    $v = \arg\min_{\{k|D_k<0\}} -\mu_k/D_k, \gamma_{\max} = -\mu_v/D_v$
    $\mu^\dagger = \mu + \gamma_{\max} D, D_d^\dagger = D_d - D_v, D_v^\dagger = 0$
    Solve the original ELMK$J^\dagger$ with the combined kernel $K = \sum_k \mu_k^\dagger K_k$
  end while
  Line search D with $\gamma \in [0, \gamma_{\max}]$
  $\mu \leftarrow \mu + \gamma D$
end while

where the stopping criteria is the variation of μ of each iteration is less than the preset threshold value or the maximum number of iterations is reached. The ultimate solution will be the kernel weight μ, and the weights w.

IV. Simulation Examples

In this section, we will give one example to substantiate the proposed prediction method based on MKELM.

The simulation is conducted in the Matlab environment running on the Windows 7 operating system, Pentium(R) Dual CPU 2.60Hz, 4 GB RAM. The root mean square error (RMSE) is used to characterize the accuracy of prediction

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(d_i - p_i)^2} \tag{10}$$

where d_i indicates the *i*-th sample of the desired output, p_i indicates the *i*-th sample of the predicted output, and n is the number of samples.

In order to demonstrate the effectiveness of our proposed prediction method, variable selection methods: max-relevance min-redundancy (mRMR) [23], is used to select the optimal variables, and extreme learning machine (ELM) [18], online extreme learning machine (OSELM) [24], support vector regression (SVR) [25], extreme learning machine with kernels (KELM) [12], and multiple kernel learning (MKL) [17] are compared in our simulation.

In this example, the proposed prediction method will be demonstrated on the benchmark Lorenz chaotic multivariate time series. The Lorenz system is as

$$\begin{cases} \frac{dx}{dt} = a(y - x) \\ \frac{dy}{dt} = (c - z)x - y \\ \frac{dz}{dt} = xy - bz \end{cases} \tag{11}$$

When $a = 10, b = 8/3, c = 28$ and $x(0) = y(0) = z(0) = 1.0$, the Lorenz system with chaotic solutions has been discovered. The fourth-order Runge-Kutta method is applied to generate a chaotic time series. The sampling time is chosen as 0.02.

Here, we use the $x(t)$, $y(t)$ and $z(t)$ together to predict the $x(t+\eta)$ time series and η is the prediction horizon. The embedded data vector is formed by 18 values of the time series

$$\begin{aligned} d(k) = \; & [x(k), x(k-\tau_1), \cdots, x(x-(m_1-1)\tau_1) \\ & y(k), y(k-\tau_2), \cdots, y(x-(m_2-1)\tau_2) \\ & z(k), z(k-\tau_3), \cdots, z(x-(m_3-1)\tau_3)]^{\mathrm{T}} \end{aligned}$$

where $m_1 = m_2 = m_3 = 6, \tau_1 = 8, \tau_2 = 7, \tau_3 = 8$.

The parameter settings are as following: The length of training set is 1500 and the length of testing set is 500. The mRMR [23] is applied to select $1 - 18$ variables, the prediction results based on the different selected variables are shown in Fig. 1. From Fig. 1, we can see that the prediction performance is different with different variables selected, and when 3 variables is selected, we get the best result. The prediction results of MKELM conducted on the Lorenz time series with 3 variable selected are shown in Fig. 2. The predicted values are fitting the actual values well, and the prediction errors are at a low level near zero.

The parameters of the compared methods are set as following. The number of hidden nodes of ELM [18], the number of hidden nodes and chunk number of OSELM [24], the regularization coefficient and kernel width of SVR [25] with Gaussian kernel, the regularization coefficient and kernel width of KELM [12] with Gaussian kernel, and the regularization coefficient of MKL [17] are chosen by 10-fold cross validation.

The prediction performance is measured by the testing error (RMSE), and all the prediction results are shown in Table I. It can be seen form Table I that, the prediction results based on selected variables are better than these based on original 18 variables, which indicate the effectiveness of the variable selection preprocessing procedure. It can also be seen form Table I that the prediction performance of MKELM is superior to the compared methods, which ensures the effectiveness of the proposed methods.

TABLE I
Comparison of Single-Step Prediction Accuracy (Lorenz-$x(t)$)

Variables	[1,7,15]	[1-18]
ELM	0.1672	0.4189
OSELM	0.0155	0.1551
SVR	0.8567	0.8848
KELM	0.0125	0.0652
MKL	0.0970	0.1056
MKELM	0.0034	0.1022

V. Conclusions

In this paper, a multivariate time series prediction method based on multiple kernel extreme learning machine is proposed. The multivariate time series is first reconstructed to phase space. Then variable selection is used to preprocess the transformed data. A prediction model named multiple kernel

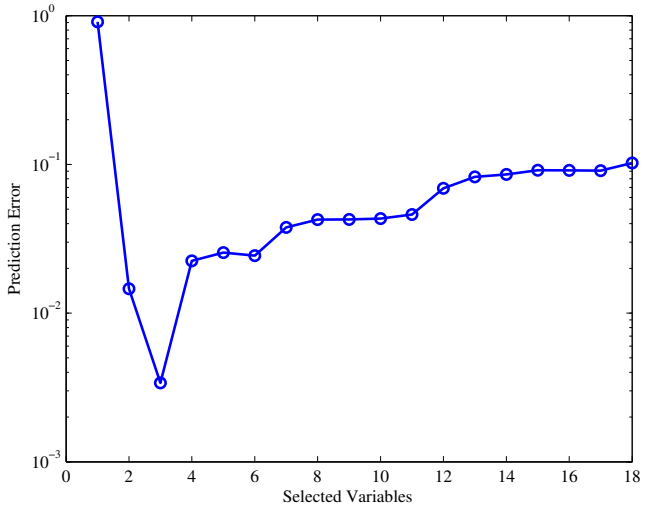

Fig. 1. Prediction Error and Selected Variables.

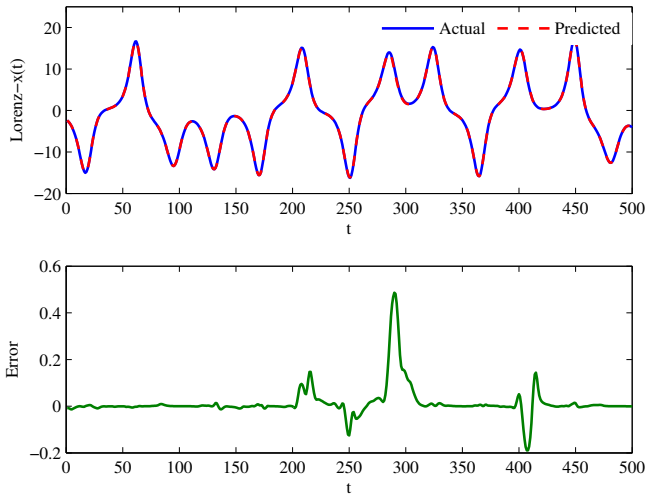

Fig. 2. Prediction results of Lorenz_$x(t)$ time series based on MKELM method.

extreme learning machine, which combines the multiple kernel learning and extreme learning machine with kernels, is proposed to model the nonlinear input-output function. The performance of the proposed method has been tested by Lorenz chaotic time series prediction simulation. The simulation results indicate that variable selection preprocessing procedure can select a compact and relevant variable set for the prediction model. Meanwhile, the proposed multiple kernel extreme learning machine outperforms other state-of-art methods.

REFERENCES

[1] P. Zhao, L. Xing, and J. Yu, "Chaotic time series prediction: From one to another," *Physics Letters A*, vol. 373, no. 25, pp. 2174–2177, 2009.

[2] N. Sapankevych and R. Sankar, "Time series prediction using support vector machines: a survey," *IEEE Computational Intelligence Magazine*, vol. 4, no. 2, pp. 24–38, 2009.

[3] L. Cao, A. Mees, and K. Judd, "Dynamics from multivariate time series," *Physica D: Nonlinear Phenomena*, vol. 121, no. 1-2, pp. 75–88, 1998.

[4] K. Chakraborty, K. Mehrotra, C. K. Mohan, and S. Ranka, "Forecasting the behavior of multivariate time series using neural networks," *Neural Networks*, vol. 5, no. 6, pp. 961–970, 1992.

[5] F. Popescu, "Robust statistics for describing causality in multivariate time series," *Journal of Machine Learning Research*, vol. 12, pp. 30–64, 2011.

[6] A. a. Jamshidi and M. J. Kirby, "Modeling multivariate time series on manifolds with skew radial basis functions," *Neural computation*, vol. 23, no. 1, pp. 97–123, Jan. 2011.

[7] F. Takens, "Detecting strange attractors in turbulence," *Dynamical systems and turbulence, Warwick 1980*, pp. 366–381, 1981.

[8] G. Huang, D. Wang, and Y. Lan, "Extreme learning machines: a survey," *International Journal of Machine Learning and Cybernetics*, vol. 2, no. 2, pp. 107–122, 2011.

[9] A. Nizar, Z. Dong, and Y. Wang, "Power utility nontechnical loss analysis with extreme learning machine method," *IEEE Transactions on Power Systems*, vol. 23, no. 3, pp. 946–955, 2008.

[10] M. Van Heeswijk, Y. Miche, T. Lindh-Knuutila, P. Hilbers, T. Honkela, E. Oja, and A. Lendasse, "Adaptive ensemble models of extreme learning machines for time series prediction," *Artificial Neural NetworksICANN 2009*, pp. 305–314, 2009.

[11] C. Lian, Z. Zeng, W. Yao, and H. Tang, "Ensemble of extreme learning machine for landslide displacement prediction based on time series analysis," *Neural Computing and Applications*, vol. 24, no. 1, pp. 99–107, 2014.

[12] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 42, no. 2, pp. 513–529, 2012.

[13] H. Jaeger, "Discovering multiscale dynamical features with hierarchical echo state networks," *Jacobs University Bremen, Tech. Rep*, 2007.

[14] X. Wang and M. Han, "Multivariate chaotic time series prediction based on hierarchic reservoirs," in *Systems, Man, and Cybernetics (SMC), 2012 IEEE International Conference on*. IEEE, 2012, pp. 384–388.

[15] A. Widodo and I. Budi, "Multi layer kernel learning for time series forecasting," in *Advanced Computer Science and Information Systems (ICACSIS), 2012 International Conference on*. IEEE, 2012, pp. 313–318.

[16] L. Duan, I. W. Tsang, and D. Xu, "Domain transfer multiple kernel learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 3, pp. 465–479, Mar. 2012.

[17] A. Rakotomamonjy, F. Bach, S. Canu, Y. Grandvalet *et al.*, "Simplemkl," *Journal of Machine Learning Research*, vol. 9, pp. 2491–2521, 2008.

[18] G. Huang, Q. Zhu, and C. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1-3, pp. 489–501, 2006.

[19] G. Feng, G. Huang, Q. Lin, and R. Gay, "Error minimized extreme learning machine with growth of hidden nodes and incremental learning," *IEEE Transactions on Neural Networks*, vol. 20, no. 8, pp. 1352–1357, 2009.

[20] Y. Miche, A. Sorjamaa, P. Bas, O. Simula, C. Jutten, and A. Lendasse, "Op-elm: optimally pruned extreme learning machine," *IEEE Transactions on Neural Networks*, vol. 21, no. 1, pp. 158–162, 2010.

[21] Z.-H. Zhou, J. Wu, and W. Tang, "Ensembling neural networks: many could be better than all," *Artificial intelligence*, vol. 137, no. 1, pp. 239–263, 2002.

[22] M. Gnen and E. Alpaydn, "Multiple kernel learning algorithms," *Journal of Machine Learning Research*, vol. 12, pp. 2211–2268, 2011.

[23] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 8, pp. 1226–1238, 2005.

[24] L. Nan-Ying, H. Guang-Bin, P. Saratchandran, and N. Sundararajan, "A fast and accurate online sequential learning algorithm for feedforward networks," *Neural Networks, IEEE Transactions on*, vol. 17, no. 6, pp. 1411–1423, 2006.

[25] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and computing*, vol. 14, no. 3, pp. 199–222, 2004.